New Packet Header Support and Key Exchange Mechanism for Secure Trivial File Transfer Protocol

Nur Nabila Mohamed, Yusnani Mohd Yussoff, Nazhatul Hafizah Kamarudin, Habibah Hashim

***Abstract*— TFTP has received a significant demand in recent years because of its lightweight feature and compatibility on limited low-level embedded system. Nevertheless, as it provides no security mechanism, questions have been raised about the safety issue especially during data collection and during system update. Despite the vulnerability matter, there is also a concern to integrate TFTP with a suitable and compatible security mechanism. Because TFTP is a lightweight protocol, the proposed security mechanism should also be simple yet powerful to protect the file transmission process against any attack. The first important method in securing the TFTP communication is to verify the new packet header that can support the key exchange operation for the next encryption process. DHKE, a general and popular key exchange mechanism usually be chosen but it is vulnerable to MITM attack. Therefore, in this work, a new DHKE-based key exchange technique is proposed to be integrated with TFTP packet header for security support which is expected to provide a secure TFTP communication in the future M2M technology.**

***Index Terms*— Diffie-Hellman Key Exchange, Machine-to-Machine, Man-In-The-Middle, Trivial File Transfer Protocol**

I. INTRODUCTION

IN M2M communication, in order to transfer data and messages effectively from or to the destination, the devices must associate with a file or data transfer protocol. Trivial File Transfer Protocol, also known as TFTP, is a simple yet multi-purpose protocol to share file, upload or download image, upgrade system, remote boot nodes and so on in many constrained/low cost embedded devices. In the last few years, the application of this protocol is considered to be irrelevant due to its reliability and security constraint. However, because of the protocol's lightweight features and its compatibility to be implemented in limited low level embedded system, TFTP application has received a significant demand until now. In the latest publication regarding TFTP in 2015 [1], the author claimed that this protocol is still massively being used in network boot/installation scenarios. This statement is strongly proved when Hitachi has introduced 'M2M Data Collection Agent' application [2][3] to collect and accumulate data from various types of sensor nodes/devices for business and industry purposes.

This paper is submitted on 26th January 2017. Accepted on 19th July 2017. The research is funded by the Fundamental Research Grant Scheme Research, Ministry of Higher Education Malaysia (600-IRMI/FRGS5/3(141/2015)). We would like to extend our acknowledgment to UniversitiTeknologi Mara (UiTM) and those who have directly or indirectly contribute to this research.

They chose to use TFTP as a lightweight communication module to collect data because they believed that its implementation can reduce the communication cost and network load. Nevertheless, questions have been raised about the safety issue during data collection and during system update. Certainly, if no security guarantee in this subject, there will be less interest in the future M2M communication paradigm. Apart from that, two major recent works have been studied that discussed the importance and necessity of security in TFTP. Research studies by [4] and [5] have presented a security solution for TFTP communication protocol for the deployment in embedded field (IoT, embedded multi-agent system etc). Further discussions about these studies will be explained in the next section in details.

In order to acknowledge the security requirements for TFTP, cryptographic method would be the best solution to secure information from being hacked during the transmission over insecure channel. Thus in this work, in order to strengthen TFTP communication protocol, a new DHKE-based key exchange technique is proposed to strengthen the protocol. After the exchange, the key obtained is utilized for next encryption and decryption process. Throughout this work, the proposed lightweight security embedding in the simple TFTP is believed to be a significant enhancement in pervasive IoT applications for managing and upgrading low-cost embedded infrastructure.

II. RELATED WORKS

A. File Transfer Protocols

Trivial File Transfer Protocol (TFTP) and File Transfer Protocol (FTP) are two general protocols used widely for sharing information. FTP is capable to transfer larger file size in TCP/IP inter networks, nevertheless, TFTP is simpler than FTP in terms of executing commands and the flexibility of using different types of transport protocol. Previous study in [6] has reported that the major difference between FTP and TFTP is that FTP establishes two connections for transferring a file for data connection and control connection. On the other hand, TFTP requires only single connection to transfer a file between client and server. Another problem is the diskless workstations issue in FTP, where some certain hardware has lack of permanent storage thus unable it to read whole TCP/IP implementation from the hard disk [7]. Meanwhile, FTP provides user authentication as it requires the client to log in with username and password, nevertheless, the username and password used in the protocol are transferred in plaintext during logging in which therefore still can be easily

recognized and unreliable. This problem thus presents a significant risk as attacker still can view the username/password to access the information obtained.

Due to the complex commands and the need to use TCP for data transport and connection, a simplified alternative of FTP protocol, TFTP, was introduces in the late of 1970s that would emphasize on small program size and simplicity over functionality [8]. Circa 1990, TFTP protocol is tremendously used in the system with limited resources because it does not require any operating system or kernel for transferring data. Nowadays, this well-known protocol is used by network administrators to upgrade router firmware and to distribute software within the corporate network as it is easy to be implemented on small amount of memory which is an important factor in the field of constrained embedded system. However, because of its simplicity, there is no access control provided which is disastrous as it allows any information to be copied to any devices without performing user validation[9].

Meanwhile, Secure File Transfer Protocol (SFTP) [7] was introduced in 2005 whereby the data transmission is executed over an encrypted connection (SSH-Tunnel). This protocol is also used to obtain general access to FTP server's file system; however, similar to FTP, it is impractical to be implemented on lightweight embedded devices regarding its complexity. Moreover, TFTP is preferable in situations when simplicity and speed are more important yet it can only be used for limited purposes because of security constraint. Therefore, this protocol would bring huge advantage in the data transmission field for embedded system if the security enhancement is improved in TFTP.

B. TFTP Implementation in Modern Technology

Previously, TFTP implementation is seen irrelevant due to the security issue. This misleading perception is proven wrong as it has been found that in recent work [2][3], a large company from Japan, Hitachi, has used TFTP as a lightweight communication module in their product known as 'M2M Data Collection Agent' to collect and accumulate data from various types of sensors/devices for the application in business and industry field. They believed that the lightweight feature of TFTP can reduce the communication cost and network load. However, the security mechanism during data collection and during system update still needs further research. According to RFC3617 [10], TFTP provides no access control mechanism to secure the protocol, as well as no protection from Man-In-The-Middle attack. Certainly, if no security guarantee in this classic issue, there will be less interest in the future M2M communication paradigm. Also, related works in Radio Frequency (RF) [11], remote attestation for Trusted Platform Modules (TPM) [12], lightweight protocol for remote management of the cloud infrastructure [13], Wide Area Network (WAN) monitoring system [14][15] shows the relevance and potential usage of TFTP implementation in recent years yet the security problem still be a primary concern as it makes the implementation vulnerable to various attacks. Hence, an integration of the proposed technique with this lightweight protocol is expected to provide minimal security property in TFTP communication.

C. Related Works in Securing TFTP

Based on previous related papers on securing TFTP in [4][5], the importance of providing security mechanism in the protocol has been realized to prevent from various attacks that will be used in low computational power devices. Nevertheless, very few studies have investigated in the field of secure TFTP protocol. So far, there are two recent works that elaborate the importance of securing TFTP. In order to enable the security support in TFTP, it is necessary to add a new packet format that is supported by TFTP option extension. Referring to related works on TFTP, we have studied the packet format that supports the security extension. The author in [4] has improved the security in TFTP communication by introducing a security extension to the general TFTP packet header using Digest Access Authentication (DAA) accompanied by hash function SHA1 for authentication credibility. This work enables the TFTP support for authentication by adding additional opcodes include GET_NONCE, RETURN <nonce>, END_SESSION, AUTH <return>. In this research work, the nonce value is used for authentication; however, researchers have not treated in much detail on testing result and discussion for the proposed authentication in the TFTP frame.

Besides, Anuar et. al [5] has introduced the security framework in TFTP by providing security negotiation before file transmission process. The author has proposed a new packet header in the option extension header format using two cryptographic principles, symmetric (AES algorithm) and asymmetric encryption (DHKE concept). This work is considered as a preliminary step on securing TFTP and still need adequate research on negotiating the key exchange process before data transmission process towards a secure TFTP communication. Another related work [16] has presented the formal proof and security reduction approach from CramerShoup encryption scheme and side channel security within fixed-time. The work in [17][18] discussed further on the security enhancement for TFTP which is also a continuation from the previous work by Anuar et. al [5]. Based on the above mentioned works regarding securing TFTP protocol, we believe that the proposed technique can provide considerable security requirement in TFTP communication.

III. PROPOSED KEY EXCHANGE PROTOCOL

In order to secure the TFTP communication protocol, it is necessary to do the key exchange operation first to share the secret key. The key exchange operation will be integrated with TFTP packet header based on the option extension format. Figure 1 explains thoroughly the DHKE operation using the communication between Client and Server as example. Client and Server agree to use p and g public integers where p is a large prime number and g is a generator of p. They choose the positive personal values, a and b, which have not been transmitted over public medium. They will then compute the public integers, based on their personal values according to

(2). They can share their public integers, *A* and *B*, over an insecure communication channel. From these public integers, a *key* can be generated by either communicating user on the basis of their own personal values where the value of the *key* turns out to be the same. This technique enables both Client and Server to generate the exact same secret *key* without transferring the *key* in physical manner but the limitation of this operation is the vulnerability to Man-In-The-Middle (MITM) attack where the adversary can intercept all the parameters passing through the public channel and sending its own public value in which finally can compromise the entire communication.

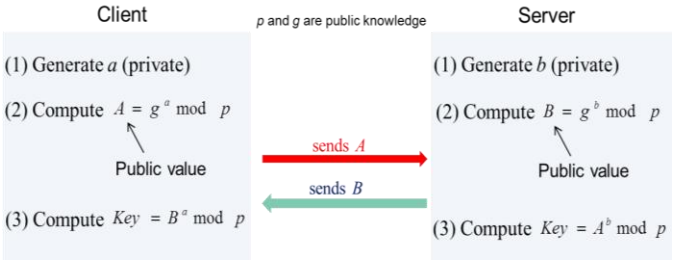

Fig. 1. DHKE general operation

To ensure a more interactive and secure communication as well as reduce the mitigation of the Man-In-The-Middle, the new modified key exchange technique is proposed to be integrated with TFTP protocol. Figure 2 shows the procedure of Client and Server communication when executing the parameters exchange process:

1. Client choose *a* (private value) and calculate $A = g^a mod\ p$
2. Client sends *A* with hash value to Server, *h(A)*.
3. Server checks the hash value, choose *b* (private value) and calculate $B = g^b\ mod\ p$. Server then compute key, $K_B = A^b\ mod\ p$. Server encrypt *B* using K_B, $C_B = E_{KB}(B)$.
4. Server sends (C_B, B) to Client.
5. Client computes $K_A = B^a\ mod\ p$, decrypts C_B using K_A to find $B' = D_{KA}(C_B)$.
6. If $B' = B$, Client then sends acknowledgement to Server.
7. Server starts to send the requested data that is encrypted with the previous secret key.

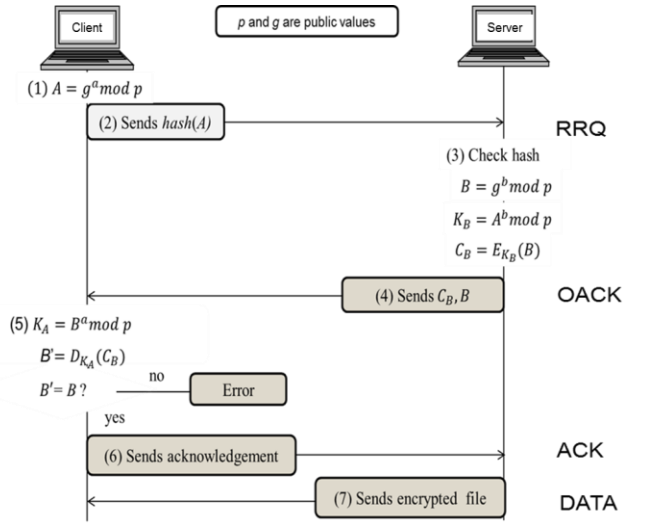

Fig. 2. Proposed key exchange model

In step (4), when the parameter (C_B, B) is sent to Client, Client would compute *B'* using her own key, and if *B'* is a similar value with *B*, Client can verify that it is interacting with the real server before sending the acknowledgement. After the key exchange and verification process, Server will then start to send the encrypted file/data.

IV. Packet Header Support

To facilitate the TFTP support for enabling the key exchange, the packet header format and its frame should be adapted according to TFTP feasibility. TFTP has an option extension feature to negotiate additional parameters to be exchanged between client and server as explained in RFC 2347 [19]. The extension allows the file transfer options to be negotiated prior to the transfer using a mechanism which is consistent with TFTP's Request Packet (RRQ or WRQ) format e.g block size, characters, additional information regarding transmission process etc [1][9]. The packet format for the extension that are appended to TFTP RRQ or WRQ packet is showed in Figure 3 below. In this work, we will use this extension to provide security negotiation in the TFTP file transfer process.

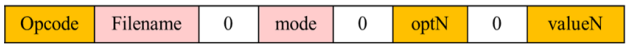

Fig. 3. TFTP Option Extension Format

Specifically, the support for RRQ_KeyEx must be implemented in the existing message format of TFTP protocol. Figure 4 shows the general format of Read Request TFTP operation meanwhile Figure 5 illustrates the proposed TFTP protocol. The option extension packet (RRQ_KeyEx) is an option to send request to Server to exchange the parameter for secret key. Value *A* packet contains Client's public parameter which is hashed with cryptographic hash function. In this communication, we intend to encrypt *B* value that is sent from Server for the next verification process at Client's side. Client will only use the accepted options where each value is associated with its option. After Client acknowledging it, Server proceeds to send the encrypted data.

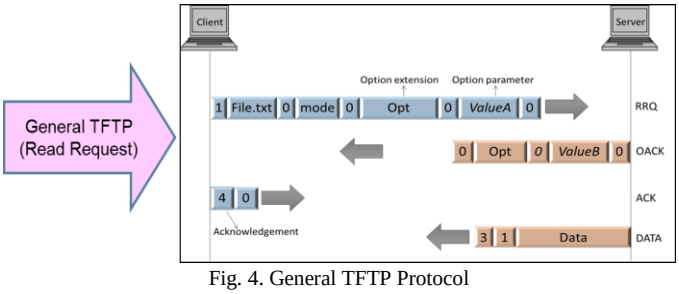

Fig. 4. General TFTP Protocol

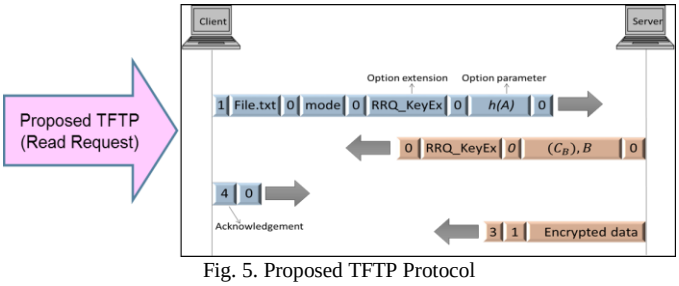

Fig. 5. Proposed TFTP Protocol

A simulation of TFTP protocol for Read Request (RRQ) packet has been performed using network simulator tool NS3 to explore the transmission when sending the header packet. As the maximum total RRQ/WRQ packet header size is 512 bytes, the designed new packet should be below than 512 bytes (4096 bit). The system runs on Laptop Intel® Core™ i3-3217U CPU @ 1.80 GHz with 4 GB of memory. The basic software environment is Linux (Ubuntu 14.04), NS3 (Release 3.26), G++ (GCC 4.8.4) and Python (2.7). Figure 6 shows the output result when the client sends 140 bytes packet header size (1024 bit parameter size) that includes the size of filename, mode of file, and the public parameters for key exchange. According to the result, the client (IP 10.1.1.1) takes time about 0.00913s to send 140 bytes packet to the server while the server (IP 10.1.1.2) takes time about 0.0085s to send acknowledgement packet to the client.

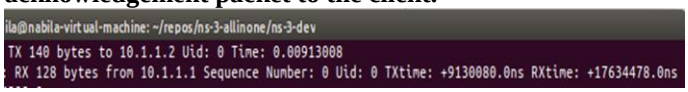

Fig. 6. TFTP RRQ Simulation

Next, simulation of various parameter size of 512, 768, and 2048 bits are also performed to explore the time difference of each values. These values are chosen based on the size of DH public parameters for key exchange, whereby all of these values are less than 512 bytes. Table 1 shows the result when using different parameters length. It can be seen that the transmission time for the Client to send RRQ header is increase when the length increase. The first length, 512 bit size is the fastest because the size is the smallest, however, 512 and 768 bit parameters size is too small and can be broken easily according to report from NIST. Meanwhile, it takes 0.00913s to send the parameters with 1024 bit length and it is considered secure as well, but 2048 bit is expected to be the safest, also its length is acceptable since the value does not exceed 512 bytes header size. Thus in the future work, real time experiment will be conducted based on this simulation results to analyze the performance of the new proposed secure TFTP protocol.

Table 1. RRQ Header Transmission Time

Size of parameters (bits)	Time (s)
512	0.00367
768	0.00785
1024	0.00913
2048	0.00989

V. CONCLUSION

As conclusion, in order to achieve the practicality to implement the protocol for transferring file on low cost constrained environment as well as protect its content, we propose a security framework as an initial findings to merge the key exchange technique based-DHKE with TFTP communication protocol**. The proposed work makes noteworthy contributions by presenting the new packet header formats that create support for securing TFTP.** It is believed that through cryptographic security implementation in TFTP using new proposed technique, information can be protected from being tampered or eavesdropped by unauthorized third party. Compared to normal TFTP which has no assurances the messages that is being sent will arrive to exact destination, this simple security solution on TFTP will satisfy the security requirements during file transmission in M2M communication technology. The proposed lightweight security mechanism is expected to provide a compatible solution in TFTP simple protocol. The enhancement will also make TFTP protocol more reliable and trustworthy to be used in commercial system especially for low cost embedded system. In the next stage of this research work, an experiment will be conducted to implement and formulate the new packet format with additional security mechanism using embedded microcontroller device.

REFERENCES

[1] P. Masotta, "TFTP Windowsize Option," 2015.
[2] "Hitachi Data Collection Agent," 2013. [Online]. Available: http://www.hitachi-solutions.co.jp/datacollection/.
[3] T. Iitsuka, "Hitachi Cloud Computing Solutions for Enterprise Information Systems," vol. 61, no. 2, pp. 53–59, 2012.
[4] G. Horvat, D. Žagar, and G. Martinović, "STFTP: Secure TFTP protocol for embedded multi-agent systems communication," *Adv. Electr. Comput. Eng.*, vol. 13, no. 2, pp. 23–32, 2013.
[5] M. A. M. Isa, N. N. Mohamed, H. Hashim, S. F. S. Adnan, J. L. A. Manan, and R. Mahmod, "A lightweight and secure TFTP protocol for smart environment," in *ISCAIE 2012 - 2012 IEEE Symposium on Computer Applications and Industrial Electronics*, 2012, no. Iscaie, pp. 302–306.
[6] D. Springall, Z. Durumeric, and J. A. Halderman, "FTP: The forgotten cloud," *Proc. - 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Networks, DSN 2016*, pp. 503–513, 2016.
[7] P. Ford-Hutchinson, "Securing FTP with TLS," pp. 1–29, 2005.
[8] K. Bollins, "The TFTP Protocol (Revision 2) RFC 1350," 1992.
[9] A. H. G. Malkin, "TFTP Blocksize Option," 1998.
[10] E. Lear, "RFC 3617 Uniform Resource Identifier (URI) Scheme and Applicability Statement for the Trivial File Transfer Protocol (TFTP)," 2003.
[11] K. F. Kao, I. E. Liao, and J. S. Lyu, "An indoor location-based service using access points as signal strength data collectors," in *2010 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010 - Conference Proceedings*, 2010, no. September, pp. 15–17.
[12] J. Schiffman, T. Moyer, T. Jaeger, and P. McDaniel, "Network-based root of trust for installation," *IEEE Secur. Priv.*, vol. 9, pp. 40–48, 2011.
[13] F. Doelitzscher, A. Sulistio, C. Reich, H. Kuijs, and D. Wolf, "Private cloud for collaboration and e-Learning services: From IaaS to SaaS," *Comput. (Vienna/New York)*, vol. 91, pp. 23–42, 2011.
[14] M. A. M. Isa, H. Hashim, J. A. Manan, R. Mahmod, M. S. Rohmad, A. H. Hamzah, M. M. A. M. Hamzah, L. Mazalan, H. Othman, and L. Adnan, "Secure System Architecture for Wide Area Surveillance Using Security, Trust and Privacy (STP) Framework," *Procedia Eng.*, vol. 41, no. Iris, pp. 480–485, 2012.
[15] J. García-Hernández and J. C. V.- Hernández, "Design Considerations for the Implementation of a Mobile IP Telephony System in a Nuclear Power Plant," *Nucl. Power - Control. Reliab. Hum. Fact ors*, 2011.
[16] Mohd Anuar Mat Isa, Habibah Hashim, Syed Farid Syed Adnan, Jamalul-lail Ab Manan, and Ramlan Mahmod, "A Secure TFTP Protocol with Security Proofs," *Proc. World Congr. Eng. 2014, (WCE 2014)*, vol. 1, pp. 3–8, 2014.
[17] N. N. Mohamed, Y. M. Yussoff, M. Anuar, M. Isa, and H. Hashim, "A Pre-shared Diffie-Hellman Key Exchange Scheme for a Secure TFTP Protocol," in *3rd International Conference on Science and Social Research*, 2016, pp. 1–13.
[18] N. N. Mohamed, Y. M. Yussoff, M. A. M. Isa, and H. Hashim, "Symmetric Encryption using Pre-Shared Public Parameters for a Secure TFTP Protocol," *J. Eng. Sci. Technol.*, vol. 12, no. 1, pp. 98–112, 2017.
[19] A. H. G. Malkin, "TFTP Option Extension (RFC 2347)," 1998.

Nur Nabila Mohamed received the Bachelor of Engineering in Electronic Information System from Shibaura Institute of Technology Japan in 2012. Then, in 2015, she received her Master in Electrical Engineering in Universiti Teknologi MARA (UiTM) Shah Alam. Currently, she is doing her PhD research in network security area in UiTM Shah Alam. Her current research is on Internet of Things, network security and lightweight security.

Yusnani Mohd Yussoff has been appointed as a Head of Computer Engineering Department in Universiti Teknologi MARA (UiTM) Shah Alam in 2015. She is currently the Deputy Dean of the Faculty of Electrical Engineering, UITM Shah Alam. Her current research includes embedded security, wireless sensor networks and trusted computing.

Nazhatul Hafizah Kamarudin received the Bachelor of Engineering in Electrical Engineering from the Stevens Institute of Technology, United States. In 2012, she received her Master of Engineering in Wireless Communication in Stevens Institute of Technology, United States. Currently, she is doing her PhD research in security and authentication area in UiTM Shah Alam. Her research work is on mobile e-health security and authentication.

Habibah Hashim is an Associate Professor in the Faculty of Electrical Engineering at Universiti Teknologi MARA (UiTM) Shah Alam and has served as the Deputy Dean of Research and Industrial Linkages between 2011 to 2014. Currently she is heading the Information Security and Trusted Information Laboratory and is pursuing her research interests in wireless and mobile networks, data communications, secure and trusted systems and Internet of Things.